DiSign

Objective

Explore authentication of documents by digitally signing a file and subsequently authenticating the document by verifying the signature.

Model

Openssl utility with the dgst command supports the options “-sign” and “-verify”. This can be used as the pattern and to verify the functioning of this utility.

User needs and requirements

Id	Need/Requirement
1	The utility shall support generating a signature file containing the digital signature of a given file.
2	Digitally signing a file should use a private key - RSA based
3	The utility shall support authenticating a file against a digital signature file using the public key of the key pair
4	The signatures shall be based on a sha256 hash of the file (32 bytes in length) at least

User wants

The following will enhance the appeal of the tool.

Id	Wants
1	The private key may be protected by a passphrase. The tool shall support passphrase protected private keys.

Specification

Command/Switch	Description
sign	Digitally sign the input file. Generates a signature file [basename of file].sig.
	Argument is the list of files. Arg1 arg2 arg3 …

Command/Switch	Description
	—private provides the name of the private key to be used in signing the file. Default is ~/.ssh/id_rsa
	—passphrase is an optional passphrase for the private key
authenticate	Verifies the signature of the file
	Argument is the list of files. arg1, arg2, ….
	—public provides the name of the public key to verify the authentication of the file. Default ~/.ssh/id_rsa.pub

Example usage

```
$ ../../../bin/disign sign lsfiles.txt
2020/01/14 12:06:33 Signing using /Users/rajasrinivasan/.ssh/
id_rsa of 1 files
2020/01/14 12:06:33 Signing lsfiles.txt creating lsfiles.txt.sig
```

```
$ ../../../bin/disign authenticate lsfiles.txt
2020/01/14 12:06:41 Authenticating using /Users/
rajasrinivasan/.ssh/id_rsa.pub of 1 files
2020/01/14 12:06:41 Verified the signature lsfiles.txt.sig of 
file lsfiles.txt
```